



Towards an Automatic Parking System using Bio-Inspired 1-D Optical Flow Sensors

Stefano Mafrica, Alain Serval, Franck Ruffier

► To cite this version:

Stefano Mafrica, Alain Serval, Franck Ruffier. Towards an Automatic Parking System using Bio-Inspired 1-D Optical Flow Sensors. IEEE International Conference on Vehicular Electronics and Safety (ICVES) 2015, Nov 2015, Yokohama, Japan. hal-01235593

HAL Id: hal-01235593

<https://hal.science/hal-01235593>

Submitted on 3 Dec 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Towards an Automatic Parking System using Bio-Inspired 1-D Optical Flow Sensors

Stefano Mafrica

PSA Peugeot Citroën, Route de Gisy
78140 Vélizy-Villacoublay, France.
Aix-Marseille Université, CNRS
ISM UMR 7287, 13288 Marseille, France.
Email: stefano.mafrica@univ-amu.fr

Alain Servel

PSA Peugeot Citroën, Route de Gisy
78140 Vélizy-Villacoublay, France.
Email: alain.servel@mps.com

Franck Ruffier

Aix-Marseille Université, CNRS
ISM UMR 7287, 13288 Marseille, France.
Email: franck.ruffier@univ-amu.fr

Abstract—Although several (semi-) automatic parking systems have been presented throughout the years [1]–[12], car manufacturers are still looking for low-cost sensors providing redundant information about the obstacles around the vehicle, as well as efficient methods of processing this information, in the hope of achieving a very high level of robustness. We therefore investigated how Local Motion Sensors (LMSs) [13], [14], comprising only of a few pixels giving 1-D optical flow (OF) measurements could be used to improve automatic parking maneuvers. For this purpose, we developed a low computational-cost method of detecting and tracking a parking spot in real time using 1-D OF measurements around the vehicle as well as the vehicle’s longitudinal velocity and steering angle. The algorithm used was composed of 5 processing steps, which will be described here in detail. In this initial report, we will first present some results obtained in a highly simplified 2-D parking simulation performed using Matlab/Simulink software, before giving some preliminary experimental results obtained with the first step in the algorithm in the case of a vehicle equipped with two 6-pixel LMSs. The results of the closed-loop simulation show that up to a certain noise level, the simulated vehicle detected and tracked the parking-spot assessment in real time. The preliminary experimental results show that the average refresh frequency obtained with the LMSs was about 2-3 times higher than that obtained with standard ultrasonic sensors and cameras, and that these LMSs therefore constitute a promising alternative basis for designing new automatic parking systems.

I. INTRODUCTION

Since the pioneering works by Paromtchik *et al.* [1] and the first commercial Intelligent Parking Assist System (IPAS) [2], a large range of (semi-) automatic parking methods have been developed over the years using all kinds of sensors, from ultrasonic sensors [3]–[5] to cameras [6]–[8], laser scanners [9], [10], and radar systems [11], [12].

On the one hand, high-performance automatic parking systems require costly sensors, such as laser scanners [9], [10] and radars [11], [12] and large computational resources to reconstruct the surrounding 3-D environment in real time during parking maneuvers.

On the other hand, most of the (semi-) automatic parking pilots available on the market involve the use of ultrasonic sensors and rear-view cameras [5], [7], [15]–[17] to detect

a free parking spot while the driver is driving along a row of parked vehicles and to autonomously control the steering system by estimating the vehicle’s ego-position via odometry.

To improve IPAS systems, recent research works and industrial developments using Around View Monitor Systems (AVMSs) have been proposed [8], [18], [19]. AVMSs mainly use wide-angle cameras (such as fish-eye and catadioptric cameras) directed toward the ground, giving a view of the surrounding environment, to detect and track parking-spots’ ground marks during parking maneuvers.

However, the autonomous vehicles of the future will require more accurate redundant information than that provided by the systems currently available on the market, mainly because of (i) the narrow field of view (FOV) and the short distance range of the ultrasonic sensors used and (ii) the low luminosity range and the low sampling frequency of standard cameras.

As far as we know, the challenge for designing the IPASs of future autonomous vehicles consists in finding low-cost, fast-response sensors providing redundant information about the obstacles present around the vehicle, combined with a sensor-data processing system requiring very low computational resources.

We therefore investigated how Local Motion Sensors (LMSs) [13], [14], giving 1-D optical flow (OF) measurements using methods based on findings obtained on the fly’s visual system [20] could be used to improve automatic parking maneuvers.

Thanks to their wide luminosity range and their sensitivity to small contrasts [14], [21], LMSs comprising only a few autoadaptive pixels seem to constitute a promising alternative to standard cameras in situations where the lighting conditions are highly variable and the visual patterns created by the vehicle’s body, for example, show small contrasts. In addition, an array of LMSs would have the following advantages with respect to ultrasonic sensors and cameras: (i) faster responses (up to 200 Hz); (ii) a custom FOV (such as 180°); (iii) a longer distance range than ultrasonic sensors; (iv) less computational cost than cameras.

In this paper, we present the first results toward developing a low computational-cost method of performing automatic parking maneuvers whereby a parking spot can be detected and tracked in real time based on a visual motion sensor setup performing 1-D OF measurements around the vehicle. The algorithm used for this purpose was composed of 5 processing steps, as described in Sec. IV.

This work was supported by CNRS Institutes (Life Science; Information and Engineering Science and Technology), Aix-Marseille University, ANRT (Association Nationale de la Recherche et de la Technologie), PSA Peugeot Citroën via the OpenLab agreement entitled “Automotive Motion Lab”.

As proof of concept, some results obtained in a simplified 2-D parking simulation implemented in Matlab/Simulink in closed loop will first be presented. Then, to report on the work in progress, we will give some preliminary experimental results corresponding to the first step in the algorithm in the case of a vehicle equipped with two 6-pixel LMSs on a vehicle.

In Sec. II, the principles underlying an elementary LMS and the visual-sensor setup used in the simulation will be described. In Sec. III, the point and line motion model used will be presented. In Sec. IV, the various steps in the algorithm we developed will be outlined. In Sec. V and VI, we will present and discuss the results of some simulations and preliminary outdoor experiments. Some conclusions will be reached in the last section.

II. LOCAL MOTION SENSORS FOR PARKING MANEUVERS

Figure 1 shows the principles underlying a 2-pixel LMS [22] and a photo of a 6-pixel LMS [13].

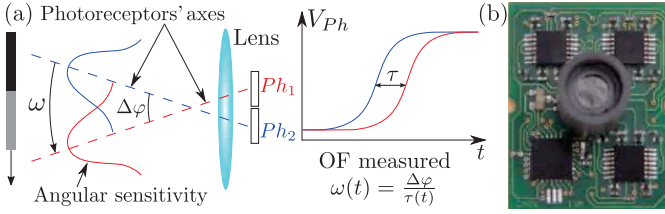


Fig. 1. (a) Principle of a 2-pixel LMS. The OF produced by a contrast moving in front of the LMS can be computed taking $\omega(t) = \frac{\Delta\varphi}{\tau(t)}$, where $\Delta\varphi$ is the interreceptor angle and τ is the time lag between the two photoreceptors' output signals [22]. (b) Photo of a 6-pixel LMS. (Adapted from [13]).

A defocused lens placed in front of 2 photoreceptors determines the interreceptor angle $\Delta\varphi$ between the 2 photoreceptors' axes and gives them a Gaussian angular sensitivity, on similar lines to what occurs in many insects' eyes. The width of the angular sensitivity determines the photoreceptors' FOV.

A visual contrast moving in front of the LMS will induce a time lag τ between the photoreceptors' output signals. By measuring this time lag, the optical flow can be computed as follows: $\omega(t) = \frac{\Delta\varphi}{\tau(t)}$ [22].

Let us take the case where one N -pixel LMS is installed at each of the four corners of the vehicle, giving up to $4(N-1)$ 1-D OF measurements all round the car. In order to obtain a 180° FOV of each LMS, N is taken to be such that $N \times \Delta\varphi = 180^\circ$, where $\Delta\varphi$ is the interreceptor angle, as shown in Fig. 1. As it is difficult in practice to obtain a 180° FOV with just one lens, there could be M N -pixel LMSs, so that $M \times N \times \Delta\varphi = 180^\circ$.

Figure 2 gives a diagram of the visual-motion sensor setup on a vehicle, in the case where 4 neighboring visual signals give 3 1-D OF measurements.

If the environment is relatively smooth, i.e. there are only small variations in the distance between the visual patterns and the LMS in the photoreceptors' FOV, we can assume that the photoreceptors will detect visual points on their axes instead of patterns in their FOV. When an OF measurement ω_i is delivered at the time \bar{t} , this will therefore be just as if a visual point has moved from the $i-1$ -th to the i -th photoreceptor's axis, and this point is now (at $t = \bar{t}$) on the i -th axis.

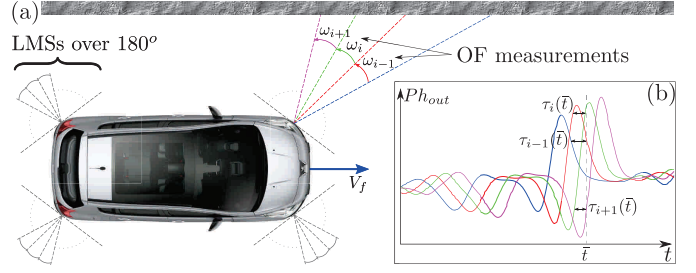


Fig. 2. (a) Diagram of the visual-motion sensor setup on a vehicle. The N -pixel LMSs are placed at the four corners of the vehicle, where N is such that $N \times \Delta\varphi = 180^\circ$. (b) An example where 4 neighboring visual signals give 3 local OF measurements.

In this preliminary study, we assumed that the neighboring visual points detected formed straight lines corresponding to the simplified 2-D profiles of the surrounding parked vehicles.

III. POINT AND LINE MOTION MODELING

As the vehicle's velocity is relatively low during parking maneuvers, we focused here on the 2-D kinematic model for a car-like vehicle moving on a plane.

Figure 3 is a top-view diagram of a car-like vehicle with a 1-D OF sensor installed on one corner.

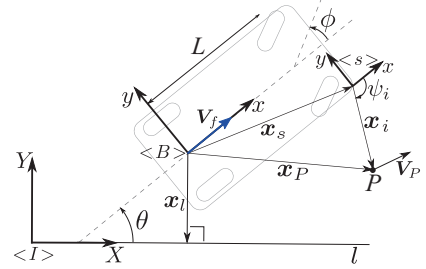


Fig. 3. Top-view diagram of a car-like vehicle with a 1-D OF sensor installed on one corner.

According to Fig. 3 and the Ackermann steering geometry for car-like vehicles, the equations for a point P moving on the same plane as the vehicle can be written as follows:

$$\begin{bmatrix} \dot{x}_i \\ \dot{y}_i \end{bmatrix} = \mathbf{f}_P(\mathbf{x}_i, \mathbf{u}) = \begin{bmatrix} \frac{(y_i + y_s) \tan \phi - L}{L} V_f + V_{Px} \\ -\frac{(x_i + x_s) \tan \phi}{L} V_f + V_{Py} \end{bmatrix} \quad (1)$$

$$\omega_i = \frac{x_i \dot{y}_i - y_i \dot{x}_i}{x_i^2 + y_i^2} \quad (2)$$

where $\mathbf{u} = [V_f \phi]^T$ are the vehicle's longitudinal velocity and its steering angle, respectively; L is the distance between the rear and front wheel axes; ${}^B \mathbf{x}_s = [x_s y_s]^T$ is the position vector of the 1-D OF sensor with respect to the body frame $\langle B \rangle$; ${}^B \mathbf{x}_P = [x_P y_P]^T$, ${}^B \mathbf{x}_i = [x_i y_i]^T$ are the position vectors of the point P with respect to the body frame $\langle B \rangle$ and the sensor frame $\langle s \rangle$, respectively; ${}^B \mathbf{V}_P = [V_{Px} V_{Py}]^T$ is the velocity vector of P with respect to the inertial frame $\langle I \rangle$; and ω_i is the angular velocity of P with respect to $\langle s \rangle$, i.e. the OF measured between the i -th and $i-1$ -th pixel of the sensor (see Sec. II). Note that the sensor frame $\langle s \rangle$ is parallel to the body frame $\langle B \rangle$ (i.e. the rotation matrix between the two frames is the identity matrix), and the

upper left index B indicates that the vectors are projected onto the body frame $\langle B \rangle$. For the sake of simplicity, this index will be dropped in what follows.

As P must lie on the i -th pixel's axis (see Sec. II), its coordinates x_i, y_i must satisfy the following equation:

$$\sin \psi_i x_i - \cos \psi_i y_i = 0, \quad (3)$$

where ψ_i is the orientation of the i -th pixel's axis, i.e. the angle known to exist between ${}^B \mathbf{x}_i$ and the x -axis of $\langle s \rangle$.

A straight line in the body frame $\langle B \rangle$ can be described by the following equation:

$$x_l x + y_l y = x_l^2 + y_l^2, \quad (4)$$

where ${}^B \mathbf{x}_l = [x_l \ y_l]^T$ is the position vector perpendicular to the line l projected onto the body frame $\langle B \rangle$.

If the line l does not move with respect to the inertial frame $\langle I \rangle$, we can assume that l is located on the X -axis of $\langle I \rangle$. In this case, in line with Fig. 3, the following equations hold:

$$\begin{bmatrix} \dot{x}_l \\ \dot{y}_l \\ \dot{\theta} \end{bmatrix} = \mathbf{f}_l(\mathbf{x}_l, \mathbf{u}) = \begin{bmatrix} \left(\frac{y_l \tan \phi}{L} - \sin^2 \theta \right) V_f \\ - \left(\frac{x_l \tan \phi}{L} - \sin \theta \cos \theta \right) V_f \\ \frac{\tan \phi}{L} V_f \end{bmatrix}, \quad (5)$$

where θ is the angle between the X -axis of $\langle I \rangle$ and the x -axis of $\langle B \rangle$.

IV. PARKING-SPOT DETECTION AND TRACKING

The algorithm presented here (see Figure 4) first computes the positions of the points detected by the LMSs using only 1-D OF measurements and the vehicle's velocity and steering angle (Step 1), and it then looks for straight lines in the clouds of points using the RANSAC method [23] (Step 2). These lines are then classified by Naive Bayes Classifiers (NBCs) (Step 3) to ensure a geometrical coherence with respect to the simplified parking-spot geometry. Lastly, the vectors perpendicular to the parking lines and the intersection points between these lines are estimated by means of Extended Kalman Filters (EKF) (Steps 4 and 5). A nonlinear tracking control is eventually performed based on the estimation of the vehicle's ego-position with respect to the parking lines and corners, but we will not focus on this process here.

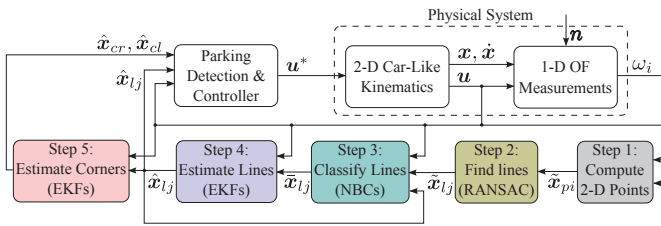


Fig. 4. Block diagram of the parking-spot detection and tracking algorithm.

Figure 5 shows an example of the results obtained in each step in the algorithm at one sampling interval during the simulation. This example involves the case of perpendicular parking, but the present algorithm can be applied to the case of parallel parking as well without making any changes.

In the simulated 2-D parking environment, the parked vehicles were simulated in the form of rectangular shapes (Fig. 5(a)) and the OF measurements of the LMSs were computed

analytically as the angular velocity of the points of intersection between the lines simulating the parked vehicles and the pixels' axes. A white Gaussian noise (\mathbf{n}) with the standard deviation σ_n is added to the positions of the intersections in order to obtain a non-linear noise in the OF measurements, which resembles the real sensors' noise more closely.

Step 1 - Computing 2-D Points:

Let us now assume that the point P in Fig. 3 does not move with respect to $\langle I \rangle$ or that its velocity components (V_{Px}, V_{Py}) are negligible with respect to the visual motion induced by the vehicle's motion, so that we can take $\mathbf{V}_P \approx \mathbf{0}$.

If we know the vehicle's longitudinal velocity and its steering angle (V_f, ϕ), it is possible by combining equation (1), (2) and (3) to compute the P coordinates as follows¹:

$$\psi_i \neq \pm \frac{\pi}{2} : \begin{cases} x_i = \frac{V_f (L \tan \psi_i - x_s \tan \phi - y_s \tan \psi_i \tan \phi)}{(\tan^2 \psi_i + 1)(L \omega_i + V_f \tan \phi)} \\ y_i = x_i \tan \psi_i \end{cases} \quad (6a)$$

$$\psi_i = \pm \frac{\pi}{2} : \begin{cases} x_i = 0 \\ y_i = \frac{V_f (L - y_s \tan \phi)}{L \omega_i + V_f \tan \phi} \end{cases} \quad (6b)$$

Figure 5(b) gives the cloud of points computed at one sampling interval during the simulation.

Step 2 - Finding Lines:

To detect a parking spot, first we want to find four straight lines fitting some subsets of the cloud of points obtained in Step 1. Then, to validate this spot, these four lines must satisfy two main conditions: they must be nearly perpendicular two by two, and the distance between the two lines in the two pairs of parallel lines must be greater than the vehicle's width and its length, respectively (see Fig. 5(a)). The validation procedure may depend on the specific parking environment (e.g. on whether it is parallel or perpendicular) and require some information from other sensors, and this part will therefore not be discussed here as it goes beyond the scope of this paper.

To look for more than one regression line in a cloud of points, we have to perform some linear clustering in order to obtain subsets containing only one line each.

In this study, we used a modified version of the RANdom SAMple Consensus (RANSAC) method [23], as it provides outliers in a statistically robust way, giving a good compromise between the level of performance and the computational cost with subsets which may be dependent but are not very numerous and are clearly separated linearly (as in the case of the 4 parking lines dealt with here). As previously done in many vision-based applications such as those designed for motion segmentation [24], the generic RANSAC algorithm is applied recursively to the subset of outliers, i.e., to all the points that do not belong to the consensus set, and whenever a line is detected, it yields the position vector $\tilde{\mathbf{x}}_l = [\tilde{x}_l \ \tilde{y}_l]^T$ satisfying equation (4). The recursion is stopped when either four lines are detected or no lines are detected in the subset of outliers, i.e., no consensus set is obtained in a maximum number of iterations n . It is worth noting that in order to maintain the high frequency provided by LMSs, we performed only a small number of iterations ($n = 100$). The robustness of the whole

¹Substituting (1) into (2) is possible because the sampling period Δt is low enough with respect to the dynamics of x_i, y_i . However, a similar result can be obtained in cases where this condition is not met, by taking a discrete approximation of (1).

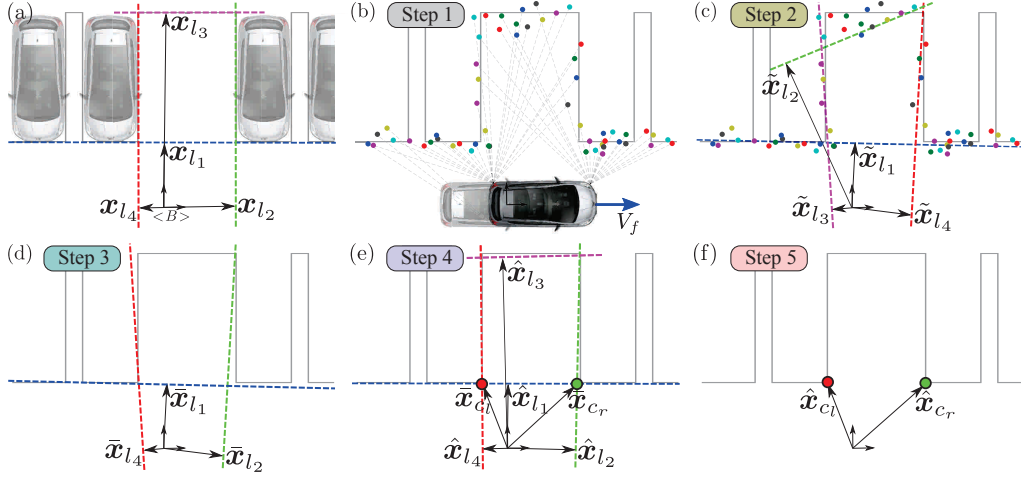


Fig. 5. Example of results obtained in each step in the parking-spot detection and tracking algorithm at one sampling interval during the simulation. (a) The simplified 2-D parking spot with the position vectors perpendicular to the 4 lines defining the spot. (b) The points detected on the pixels' axes, the positions of which were computed using the 1-D OF measurements delivered by the LMSs placed at the vehicle's corners (Step 1). A white Gaussian noise is added to the positions of these points in order to add a non-linear noise to the OF measurements. (c) The position vectors of the lines obtained by RANSAC in the cloud of points (Step 2). (d) The position vectors of the lines classified by NBCs (Step 3). (e) The position vectors of the lines estimated by EKFs (Step 4). A first approximation of the parking corners' position is given by the points of intersection between each pair of perpendicular lines. (f) The position vectors of the parking corners estimated by EKFs (Step 5).

algorithm is in fact increased by applying optimal classification (Step 3) and estimation (Step 4,5) methods to the straight lines obtained.

At each sampling interval, the h -th line \tilde{x}_{l_h} given by the algorithm can correspond to any of the four lines defining the parking spot ($x_{l_1 \dots l_4}$), or even to none of them.

Figure 5(c) shows the four lines detected by RANSAC in the cloud of points presented in Fig. 5(b). Note that \tilde{x}_{l_2} (the green line in Fig. 5(c)) differs from the corresponding line x_{l_3} (the pink line in Fig. 5(a)), while $\tilde{x}_{l_3}, \tilde{x}_{l_4}$ (the red and pink lines in Fig. 5(c)) are not in the right order as they correspond to x_{l_2}, x_{l_4} (the green and red lines in Fig. 5(a)).

Step 3 - Classifying Lines:

Let us take the four straight lines $x_{l_1 \dots l_4}$ defining the parking spot (Fig. 5(a)) and consider that they belong to four classes $\mathcal{L}_{1 \dots 4}$.

To determine whether the h -th line \tilde{x}_{l_h} detected in Step 2 belongs to the j -th class \mathcal{L}_j , we used Naive Bayes Classifiers (NBCs), giving normal and uncorrelated probability distributions for the two components of \tilde{x}_{l_h} . We therefore computed the Bayes probability as follows²:

$$\begin{cases} p_{j,h} = p(\mathcal{L}_j | x_{l_h}, y_{l_h}) = \frac{p(\mathcal{L}_j)p(x_{l_h}|\mathcal{L}_j)p(y_{l_h}|\mathcal{L}_j)}{p(x_{l_h}, y_{l_h})} \\ p(x_{l_h}|\mathcal{L}_j) = \frac{1}{\sqrt{2\pi\sigma_{x_j}^2}} e^{-\frac{(x_{l_h} - \mu_{x_j})^2}{2\sigma_{x_j}^2}} \quad (\text{same for } y_{l_h}) \\ p(x_{l_h}, y_{l_h}) = \sum_{j=1}^4 p(\mathcal{L}_j)p(x_{l_h}|\mathcal{L}_j)p(y_{l_h}|\mathcal{L}_j) \end{cases} \quad (7)$$

where we considered $p(\mathcal{L}_j) = \frac{1}{4}$, $[\mu_{x_j} \ \mu_{y_j}]^T = \hat{x}_{l_j}^-(k)$ and $[\sigma_{x_j} \ \sigma_{y_j}]^T = \mathbf{f}_l(\hat{x}_{l_j}(k-1), \mathbf{u}(k-1))$, with k denoting the k -th sampling period (i.e., $t = k\Delta t$), and $\hat{x}_{l_j} = [\hat{x}_{l_j} \ \hat{y}_{l_j}]^T$ being

the position vector of j -th line estimated by the EKF and $\hat{x}_{l_j}^-$ its "a priori" estimate (see the next subsection for details).

We classified the line \tilde{x}_{l_h} in the class \mathcal{L}_j if $p_{j,h} = \max_{j,h} p_{j,h}$ and $p_{j,h} > 0.95$. To avoid any misunderstandings, we specified the j -th classified line $\tilde{x}_{l_j} = [\tilde{x}_{l_j} \ \tilde{y}_{l_j}]^T$.

Figure 5(d) shows the three lines which were classified out of the four lines in Fig. 5(c). In particular, \tilde{x}_{l_2} (the green line in Fig. 5(c)) was dropped because it was too different from the corresponding line x_{l_3} (the pink line in Fig. 5(a)), while $\tilde{x}_{l_3}, \tilde{x}_{l_4}$ (the red and pink lines in Fig. 5(c)) were classified as $\tilde{x}_{l_4}, \tilde{x}_{l_2}$ (the green and red lines in Fig. 5(a)).

Step 4 - Estimating Lines:

In order to obtain a robust continuous estimation of the parking line vectors, an Extended Kalman Filter (EKF) based on a discrete approximation for the model presented in (5) was implemented for each of the four lines $x_{l_1 \dots l_4}$ by taking the lines classified in Step 3 ($\tilde{x}_{l_1 \dots l_4}$) to be actual measurements.

For each line, we took the first-order discrete approximation for the model presented in (5) as follows:

$$\begin{cases} x_{l_k} = \hat{\mathbf{f}}_l(x_{l_{k-1}}, \mathbf{u}_{k-1}, \mathbf{w}_{l_{k-1}}) \\ = (\mathbf{f}_l(x_{l_{k-1}}, \mathbf{u}_{k-1}) + \mathbf{w}_{l_{k-1}})\Delta t + x_{l_{k-1}} \\ z_{l_k} = \mathbf{h}_l(x_{l_k}, \mathbf{v}_{l_k}) = x_{l_k} \end{cases} \quad (8)$$

where the index k denotes the k -th sampling period (i.e., $t = k\Delta t$); $\mathbf{w}_l, \mathbf{v}_l$ denotes the model and the measurement noise, respectively, and they are assumed to be independent white noises and to have normal probability distributions, i.e. $p(\mathbf{w}) \sim N(0, Q)$ and $p(\mathbf{v}) \sim N(0, R)$, where Q, R are the covariance matrices which, in the present case, are diagonal³.

At each sampling period, the steps involved in each EKF can be summarized as follows:

²The hypothesis that the two components of \tilde{x}_{l_h} had normal and uncorrelated probability distributions was adopted on the basis of what was observed statistically during several simulations with trajectories of various kinds.

³The hypothesis that the components of the model and the measurement noise had normal and uncorrelated distributions was adopted on the basis of what was observed statistically during several simulations with trajectories of various kinds.

- 1) $\hat{\mathbf{x}}_{l_k}^- = \hat{\mathbf{f}}_l(\hat{\mathbf{x}}_{l_{k-1}}, \mathbf{u}_{k-1}, \mathbf{0})$
- 2) $P_k^- = F_k P_{k-1} F_k^T + W_k Q W_k^T$
- 3) $K_k = P_k^- H_k^T (H_k P_k^- H_k^T + V_k R V_k^T)^{-1}$
- 4) $\hat{\mathbf{x}}_{l_k} = \hat{\mathbf{x}}_{l_k}^- + K_k (z_{l_k} - \mathbf{h}_l(\hat{\mathbf{x}}_{l_k}^-, \mathbf{0}))$
- 5) $P_k = (I - K_k H_k) P_k^-$

where $F = \frac{\partial \hat{\mathbf{f}}_l}{\partial \mathbf{x}_l}$, $H = \frac{\partial \mathbf{h}_l}{\partial \mathbf{x}_l}$, $W = \frac{\partial \hat{\mathbf{f}}_l}{\partial \mathbf{w}_l}$, $V = \frac{\partial \mathbf{h}_l}{\partial \mathbf{v}_l}$ are the Jacobian matrices; P, P^- are the error covariance matrix and its “a priori” estimate, respectively; and I is the identity matrix.

In the present study, the j -th EKF was enabled when the j -th line was detected and validated for the first time on the basis of the time correlation in the values of $\hat{\mathbf{x}}_{l_j}$ (hypothesis test), but in practice, this could be done using other sensors such as ultrasonic sensors, cameras or radars installed on the vehicle.

The initial estimate of $\hat{\mathbf{x}}_{l_j}$ was set at the first value of $\hat{\mathbf{x}}_{l_j}$ validated, while the initial estimate of P_k was set at the identity matrix.

When there was no measurement of the j -th EKF, i.e. no $\hat{\mathbf{x}}_{l_j}$ in Step 3, the Kalman gain K_k was set at zero, so that it was still possible to have an estimation of \mathbf{x}_{l_j} based on the “a priori” prediction $\hat{\mathbf{x}}_{l_j}^-$.

Figure 5(e) shows the four lines estimated by the four EKFs. Note that although there was no measurement corresponding to $\hat{\mathbf{x}}_{l_3}$ in Fig. 5(d), we still had an estimation of $\hat{\mathbf{x}}_{l_3}$.

Step 5 - Estimating Corners:

The position vectors of the four parking corners can be estimated in terms of the intersection point between each pair of perpendicular lines estimated in Step 4. This makes it possible to compute the j -th corner vector $\hat{\mathbf{x}}_{c_j} = [\hat{x}_{c_j} \hat{y}_{c_j}]^T$ by substituting $\hat{\mathbf{x}}_{l_j}$ and $\hat{\mathbf{x}}_{l_{j+1}}$ into (4) and combining the two resulting equations. Here we dealt only with the two outer corners (the green and red circles in Fig. 5(e)). We took $\hat{\mathbf{x}}_{c_l}, \hat{\mathbf{x}}_{c_r}$ to specify the position vectors of the left and right corners, respectively.

This estimate of the corners’ positions was not accurate enough to be able to perform a feedback tracking control on it. In particular, when the car is crossing the lines, i. e. $\mathbf{x}_l \approx \mathbf{0}$, a very small error in $\hat{\mathbf{x}}_l$ can lead to a very large error in $\hat{\mathbf{x}}_c$.

In order to obtain a robust continuous estimation of the corner vectors, an EKF based on a discrete approximation of the model presented in (1) was implemented for each of the two outer corners ($\mathbf{x}_{c_l}, \mathbf{x}_{c_r}$), by taking the line intersection points defined above ($\hat{\mathbf{x}}_{c_l}, \hat{\mathbf{x}}_{c_r}$) to be actual measurements.

The same procedure as that applied with the EKF on \mathbf{x}_l in the previous subsection can be used with the EKF on \mathbf{x}_c (taking the same model and the same measurement noise). We only had to replace the index l by the index c and take the function $\mathbf{f}_P(\mathbf{x}_c, \mathbf{u})$ defined in (1) instead of $\mathbf{f}_l(\mathbf{x}_l, \mathbf{u})$.

The EKFs for \mathbf{x}_{c_l} and \mathbf{x}_{c_r} were enabled when the EKFs for $\mathbf{x}_{l_1}, \mathbf{x}_{l_2}$ and $\mathbf{x}_{l_1}, \mathbf{x}_{l_4}$ were enabled, respectively (the blue and green lines and the blue and red lines, respectively, in Fig. 5(e)).

The initial estimates for $\hat{\mathbf{x}}_{c_l}$ and $\hat{\mathbf{x}}_{c_r}$ were set at the first value of the intersection point between $\hat{\mathbf{x}}_{l_1}, \hat{\mathbf{x}}_{l_2}$ and $\hat{\mathbf{x}}_{l_1}, \hat{\mathbf{x}}_{l_4}$, respectively, while the initial estimate of P_k was set at the identity matrix.

Note that although we defined the parking spot by the above four lines $\mathbf{x}_{l_1 \dots 4}$, the parking spot can be detected

even if \mathbf{x}_{l_3} is not found (i.e. in the case of an “open” or “homogeneous” background), and depending on the automatic pilot’s requirements, a tracking control of the ego-vehicle’s trajectory can eventually be performed only on the basis of the two outer corners’ positions ($\mathbf{x}_{c_l}, \mathbf{x}_{c_r}$) and the two lateral lines’ orientations ($\mathbf{x}_{l_2}, \mathbf{x}_{l_4}$). For safety reasons, information about the parking-spot depth (i.e. the distance between the vehicle and \mathbf{x}_{l_3}) should be provided by means of other sensors (e.g. ultrasonic sensors or cameras).

V. CLOSED-LOOP SIMULATION RESULTS

In the simulation presented here, we used 40-pixel LMSs with $\Delta\varphi = 4.5^\circ$ ($N \times \Delta\varphi = 40 \times 4.5^\circ = 180^\circ$) and a white Gaussian noise with standard deviation $\sigma_n = 10^{-2}$. The algorithm was run at 100 Hz, giving a sampling period of $\Delta t = 10$ ms. We decided to simulate the case of perpendicular parking because it involves more complex maneuvers than parallel parking, which gives noisier OF measurements than the parallel case. In addition, to obtain results that are closer to the real case, we took into account only the OF measurements in $[-350^\circ, -1^\circ]$ and $[1^\circ, 350^\circ]$ (see [13], [14]).

For the simulated results presented here, the stages of the parking controller can be summarized as follows:

Parking stage 1 - V_f constant, ϕ in closed loop:

Drive at constant velocity ($V_f = 1 \frac{m}{s}$) parallel to the first parking line \mathbf{x}_{l_1} (blue line in Fig. 5) until the two outer corners $\mathbf{x}_{c_l}, \mathbf{x}_{c_r}$ are found: if the distance between \mathbf{x}_{c_l} and \mathbf{x}_{c_r} is greater than a minimum value of the parking-spot width then go to the next step, otherwise continue to look for a parking spot. To drive parallel to \mathbf{x}_{l_1} , the steering angle was computed as $\phi = \arctan(k_\phi L \frac{\theta^* - \hat{\theta}_{l_1}}{V_f})$, where $\hat{\theta}_{l_1}$ is the vehicle’s orientation with respect to \mathbf{x}_{l_1} estimated by the first EKF (see equation (8)), $\theta^* = \frac{\pi}{2}$ and $L = 2$ m.

Parking stage 2 - V_f and ϕ constant:

Drive away from the parking spot at constant velocity and steering angle ($V_f = 1 \frac{m}{s}$, $\phi = -\frac{\pi}{9}$) for a fixed time in order to bring the vehicle to a suitable pose for facilitating the backwards maneuver. Then, compute the reference trajectory \mathbf{X}_c^* of the vehicle’s ego-position with respect to the inertial frame $\langle I \rangle$ centered to one of the two corners. The trajectory from the actual pose to the desired pose, i.e. being in the middle between \mathbf{x}_{c_l} and \mathbf{x}_{c_r} and parallel to \mathbf{x}_{l_2} (or \mathbf{x}_{l_4}), was computed as a third-order polynomial function using the boundary conditions on the initial and final position and velocity.

Parking stage 3 - V_f and ϕ in closed loop:

Drive backwards following the reference trajectory by applying a feedback linearizing control based on the Ackermann model as follows [25]: $V_f = -\sqrt{\dot{\mathbf{X}}_c^2 + \dot{\mathbf{Y}}_c^2}$ and $\phi = \arctan(L \frac{\dot{\mathbf{X}}_c \ddot{\mathbf{Y}}_c - \dot{\mathbf{Y}}_c \ddot{\mathbf{X}}_c}{V_f^3})$, where $\dot{\mathbf{X}}_c = [\dot{\mathbf{X}}_c \dot{\mathbf{Y}}_c]^T = \dot{\mathbf{X}}_c^* + k_1(\mathbf{X}_c^* - \hat{\mathbf{X}}_{c_l})$ and $\ddot{\mathbf{X}}_c = [\ddot{\mathbf{X}}_c \ddot{\mathbf{Y}}_c]^T = \ddot{\mathbf{X}}_c^* + k_2(\dot{\mathbf{X}}_c^* - \dot{\hat{\mathbf{X}}}_{c_l})$, $\hat{\mathbf{X}}_{c_l}$ being the estimation on the vehicle’s ego-position with respect to the left corner, i.e. $\hat{\mathbf{X}}_{c_l} = -{}^I R_B \hat{\mathbf{x}}_{c_l}$, and ${}^I R_B$ being the rotation matrix from $\langle B \rangle$ to $\langle I \rangle$.⁴

⁴We used the estimation $\hat{\mathbf{x}}_{c_l}$ (left corner) since the estimation $\hat{\mathbf{x}}_{c_r}$ (right corner) was not enough reliable during the first part of the maneuver because we have no measurements for \mathbf{x}_{l_2} due to the occlusion.

Parking stage 4 - V_f constant, ϕ in closed loop:

Drive backwards at low constant velocity ($V_f = -0.5 \frac{m}{s}$) parallel to x_{l_2} (or x_{l_4}) by computing the steering angle as $\phi = \arctan(k_\phi L \frac{\theta^* - \hat{\theta}_{l_2}}{V_f})$, with $\theta^* = -\frac{\pi}{2}$, until the parking maneuver is done.⁵

Figure 6 shows the errors in the line and corner position vectors which occurred in Steps 2 to 5, as described in Sec. IV, during a given trajectory of the simulated vehicle.

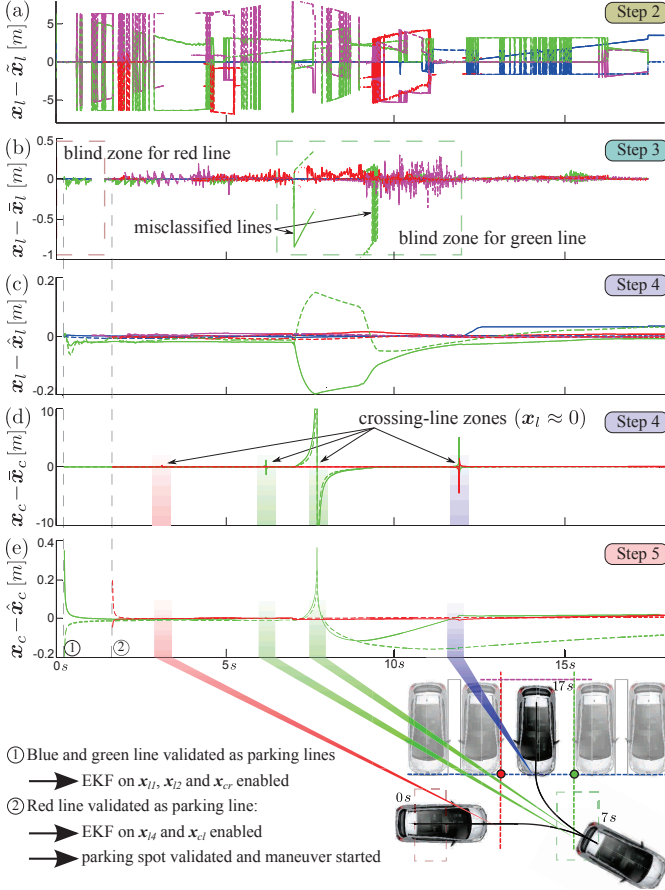


Fig. 6. Errors in the line and corner position vectors occurring during a given trajectory of the simulated vehicle at: (a) Step 2 ($x_l - \hat{x}_l$); (b) Step 3 ($x_l - \hat{x}_l$); (c) Step 4 ($x_l - \hat{x}_l$); (d) Step 4 ($x_c - \hat{x}_c$); (e) Step 5 ($x_c - \hat{x}_c$). The colors in this figure correspond to those of the lines and corners presented in Fig. 5. The continuous and dotted curves correspond to the x and y coordinates of the vectors, respectively.

In Step 2, the errors between the real lines and the lines obtained with RANSAC ($x_l - \hat{x}_l$) were sometimes relatively large (Fig. 6(a)). As explained above, this was due to the fact that the lines given by RANSAC were sometimes not in the right order (for instance, the second line obtained occasionally corresponded to the third one, as in Fig. 5(c)), or they were sometimes spurious lines.

In Step 3, the errors between the real lines and the lines classified by NBCs ($x_l - \hat{x}_l$) were much smaller, but they could still be relatively large, depending on the noise (Fig. 6(b)).

⁵The gains $k_\phi, k_1, k_2 > 0$ were dynamically adapted in order to avoid any saturation of the control inputs.

In Step 4, the errors between the real lines and the lines estimated by the EKFs ($x_l - \hat{x}_l$) were about one order of magnitude smaller, and an estimation was still delivered even when no corresponding line was found, i.e. no measurement \bar{x}_l was obtained (Fig. 6(c)). However, when the car was crossing the lines, i.e. $x_l \approx 0$, a very small error in \hat{x}_l could lead to a very large error in \hat{x}_c (corresponding to the shaded regions in Fig. 6(d)).

In Step 5, the errors between the real corners and those estimated by the EKFs ($x_c - \hat{x}_c$) were much smaller, especially when crossing lines (Fig. 6(e)).

The preliminary controller and the results presented here could be further improved, for instance by looking for the right corner's position x_{c_r} in the cloud of points while the vehicle is in the blind zone for the x_{l_2} (see Fig. 6) in order to improve the reliability on the estimation \hat{x}_{c_r} and perform a closed-loop maneuver based on the estimation of both corners' positions.

VI. PRELIMINARY EXPERIMENTAL TESTS

Preliminary tests were carried out on a real vehicle (*Peugeot 3008*) in order to check the validity of the first step in the algorithm presented in Sec. IV. In particular, two 6-pixel LMSs ($\Delta\varphi \approx 1.5^\circ$ [13] and 3.8° [14]) and a webcam (*Logitech B905*) were tied to a supporting slab which was attached to the front fender, as shown in Fig. 7(a). The slab could be moved and rotated along the z -axis in order to adjust the sensors' height and their orientation.

Figure 7(b) shows an example of the OF measured by one of the two 6-pixel LMSs presented in Fig. 7(a) ($\Delta\varphi \approx 3.8^\circ$ [14]) when the vehicle was moving alongside two parallel parked vehicles. Figure 7(c) shows the vehicle's trajectory, along with the positions of the 2-D points computed by applying equation (6) to the OF measurements in Fig. 7(b).

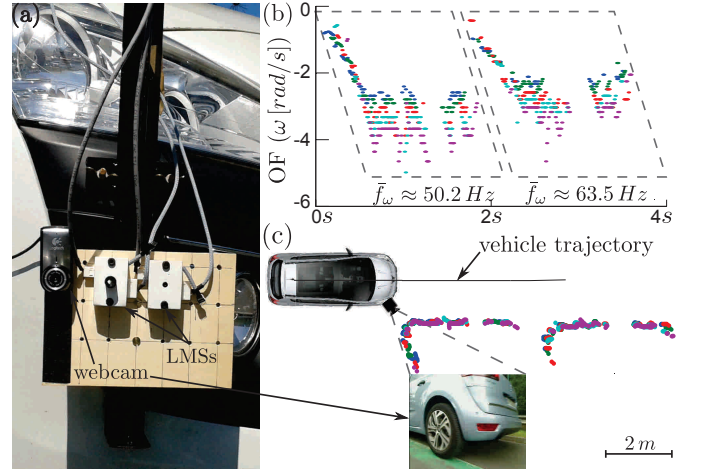


Fig. 7. (a) A picture of the sensor's supporting slab attached the front fender with the two 6-pixel LMSs and the webcam. (b) Example of the OF measured by one of the two 6-pixel LMSs presented in Fig. 7(a) ($\Delta\varphi \approx 3.8^\circ$ [14]) when the vehicle was moving alongside two parallel parked vehicles. The dotted lines indicate the OFs produced by each of the two parked vehicles, and \bar{f}_w indicates the average refresh frequency of the OF measurements in each dotted-line region. (c) The simulated vehicle's trajectory, along with the positions of the 2-D points computed by applying equation (6) to the OF measurements presented in Fig. 7(b). The picture in the lower part of the figure is the webcam image taken when the vehicle was in the position shown here (webcam's FOV $\sim 45^\circ$).

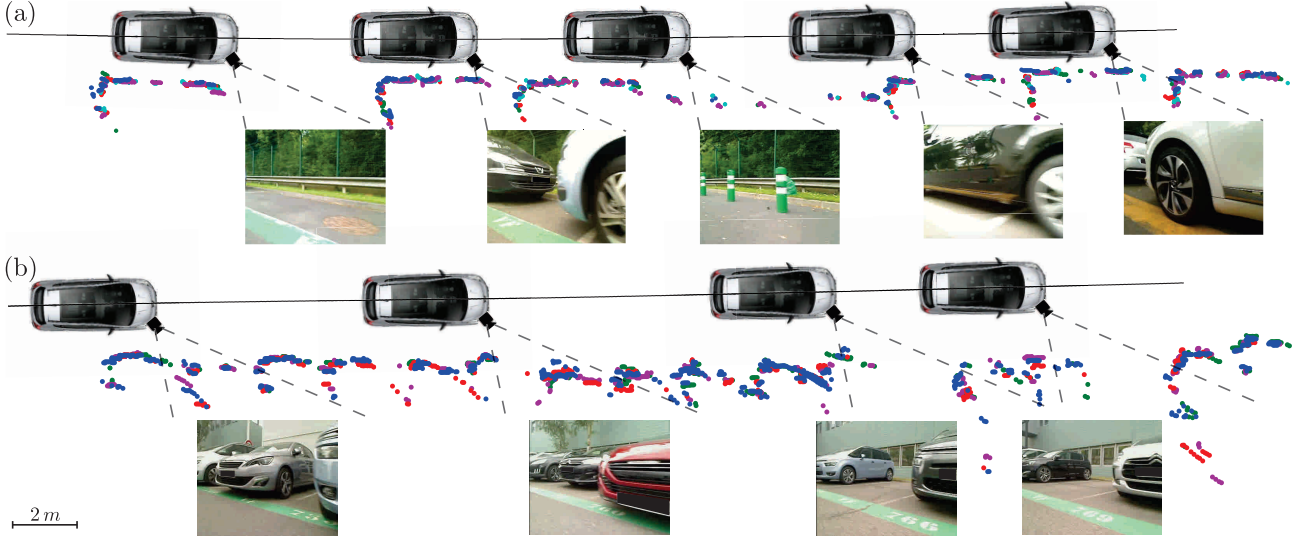


Fig. 8. Example of vehicle's trajectory along with the positions of the 2-D points computed using the OF measured by one of the two 6-pixel LMSs presented in Fig. 7(a) ($\Delta\varphi \approx 3.8^\circ$ [14]) when the vehicle was moving along a (a) parallel and (b) a perpendicular parking lane at a longitudinal velocity ranging from 3 to $9 \frac{m}{s}$. The pictures in the lower part of each subfigure are the webcam images taken when the vehicle was in the positions presented in the top part, and the dotted lines give a rough indication of the webcam's FOV ($\sim 45^\circ$).

To compute the OF measurements ω_i , the time lag τ_i between the i -th and $i-1$ -th pixel output signal (see Fig. 1, 2) was estimated using a cross-correlation method inspired by the Reichardt-Hassenstein model [26]. First we delayed one of the two signals by a time $\tau_k = k\Delta t$ in a fixed time window, and we then computed the cross-correlation between the delayed and non-delayed signals. We then set τ_i at a value equal to the time lag τ_k giving the maximum cross-correlation, as long as this maximum was greater than 0.99. As the precision in τ_i is constant and ω_i is inversely proportional to τ_i , the precision in ω_i is proportional to ω_i itself, as we can notice by looking at Fig. 7(b).

The data acquisition and the data processing were performed on a laptop PC (*Intel Core i7-2620M CPU 2.70 GHz*) with RTMaps software. In particular, the pixel output signals delivered by the LMSs were sampled at a rate of $1 kHz$ by the embedded micro-controller (*Microchip dsPIC33FJ128GP802*) and transmitted to the PC via serial communication, whereas the vehicle's longitudinal velocity and its steering angle were acquired at $20 Hz$ by the PC via the vehicle's CAN bus. The OF and the positions of the 2-D points were computed at a frequency of $100 Hz$.

Figure 8 shows two examples of the vehicle's trajectory along with the positions of the 2-D points computed using the OF measured by one of the two 6-pixel LMSs presented in Fig. 7(a) ($\Delta\varphi \approx 3.8^\circ$ [14]) when the vehicle was moving along a parallel (Fig. 8(a)) and a perpendicular parking lane (Fig. 8(b)) at a longitudinal velocity ranging from 3 to $9 \frac{m}{s}$.

The refresh frequency of the OF measurements, that is to say the ratio between the number of measurements performed in a given time interval and the time interval itself, can vary significantly, depending on the empty spaces and the contrasts provided by the car bodies (see the pictures in the lower part of Fig. 8(a) and (b)), giving both dense and sparse clouds of points. However, the average refresh frequency \bar{f}_ω for the OF produced by the parked vehicles ranged from 30 to $65 Hz$, which is about 2-3 times greater than those of standard

ultrasonic sensors and cameras.

A few 2-D points were generated in the center of Fig. 8(a) as the vehicle was moving along an empty parking place lined with small traffic poles (see the third picture in Fig. 8(a)). No measurements were made here on the background of the spots because it was too far from the LMS. In fact, the analog band-pass filter implemented on the LMS (see [14]) cuts off the low-frequency variations in the photoreceptors' signals produced by the slow visual motion of the background. This issue can be fixed by adjusting the analog filtering part of the LMS, but this has not been shown here.

It is worth noting that in the case of both parallel and perpendicular parking, it is possible to recognize the shapes of the parked vehicles in the clouds of points, even though few measurements are sometimes made due to the reflections on the car bodies (see, for instance, the fourth picture in Fig. 8(a)).

In the case of perpendicular parking, the measurements were noisier than in the parallel case, mainly because of (i) the occlusions of the cars' sides and (ii) the misalignment of the cars (see for instance the first and second pictures in Fig. 7(b)).

The qualitative results presented here validate the first step in the algorithm we have developed. Although it was not possible, in the case of this example, to apply the other steps in the algorithm because there were too few points in each sampling interval, it seems likely that by using LMSs with a larger number of pixels (or several 6-pixel LMSs) delivering 1-D OF measurements all round the vehicle, we will probably have enough points to be able to apply all the steps.

VII. CONCLUSIONS

In this paper, we have presented a low computational-cost method for detecting and tracking a parking spot in real time based on a visual motion sensor setup performing 1-D OF measurements around the vehicle.

The advantages of this method can be summed up as follows:

- the average refresh frequency is about 2-3 times higher than with standard ultrasonic sensors and cameras;
- it produces redundant information about a parking spot, in cases where ultrasonic sensors and cameras are liable to be inaccurate (e.g., when the parked vehicles are too far apart or too close, under wide-range lighting conditions and when indistinct visual patterns are visible on the car bodies);
- it can detect a candidate parking spot before passing it, thanks to the long-distance, wide-angle range of view;
- it can track the parking spot in real time, thanks to the wide-angle range of view, as well as the high-frequency measurements and low computational-cost processing;
- it makes it possible to eventually apply a nonlinear tracking control strategy using the estimation of the vehicle's ego-position with respect to the parked vehicles.

For these reasons, LMSs can provide a good additional information to automatic parking systems as well as to other automotive applications. In particular, by using LMSs with several pixels (or several few-pixel LMSs) giving a 180° FOV at each of the car's four corners, it should be possible to apply the present method in real time to performing safe closed-loop parking maneuvers.

Although we focused here only on the information provided by the LMSs, data provided by other sensors (such as ultrasonic sensors) could be integrated into the algorithm to achieve more robust performances.

Studies to improve the closed-loop parking maneuvers and experimental tests with several LMSs on a real car-like robot are now under way.

ACKNOWLEDGMENTS

We are most grateful to M. Boyron and J. Dipéri for their involvement in the electronic and mechanical design of the sensors, and to S. Prodhomme and J. Masson for their help with the sensor support on the vehicle. We also thank S. Allano for his assistance.

REFERENCES

- [1] I. E. Paromtchik and C. Laugier, "Autonomous parallel parking of a nonholonomic vehicle," in *Proc. IEEE Intell. Veh. Symp.*, 1996.
- [2] S. Hiramatsu, "Rearview camera based parking assist system with voice guidance," SAE Paper No: 2002-01-0759, 2002.
- [3] P. Degerman, J. Pohl, and M. Sethson, "Hough transform for parking space estimation using long range ultrasonic sensors," SAE Paper No: 2006-01-0810, 2006.
- [4] W.-J. Park, B.-S. Kim, D.-E. Seo, D.-S. Kim, and K.-H. Lee, "Parking space detection using ultrasonic sensor in parking assistance system," in *Proc. IEEE Intell. Veh. Symp.*, 2008.
- [5] T.-h. Hsu, J.-f. Liu, P.-n. Yu, W.-s. Lee, and H. J.-s., "Development of an automatic parking system for vehicle," in *Proc. IEEE Veh. Power Prop. Conf.*, 2008.
- [6] K. Fintzel and R. Bendahan, "3D parking assistant system," in *Proc. IEEE Intell. Veh. Symp.*, 2004.
- [7] J. K. Suhr, H. G. Jung, K. Bae, and J. Kim, "Automatic free parking space detection by using motion stereo-based 3D reconstruction," *Machine Vision and Applications*, vol. 21, pp. 163–176, 2010.
- [8] S.-e. Shih, W.-h. Tsai, and S. Member, "A Convenient Vision-Based System for Automatic Detection of Parking Spaces in Indoor Parking Lots Using Wide-Angle Cameras," *IEEE Trans. Intell. Transp. Syst.*, vol. 63, pp. 2521–2532, 2014.
- [9] M. Montemerlo, J. Becker, S. Bhat, H. Dahlkamp, D. Dolgov, S. Ettinger, D. Haehnel, T. Hilden, G. Hoffmann, B. Huhnke, D. Johnston, S. Klumpp, D. Langer, A. Levandowski, J. Levinson, J. Marcil, D. Orenstein, J. Paefgen, I. Penny, A. Petrovskaya, M. Pflueger, G. Stanek, D. Stavens, A. Vogt, and S. Thrun, "Junior: The Stanford Entry in the Urban Challenge," *J. Field Robot.*, vol. 25, pp. 569–597, 2008.
- [10] P. Jeevan, F. Harchut, B. Mueller-Bessler, and B. Huhnke, "Realizing Autonomous Valet Parking with automotive grade sensors," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2010.
- [11] S. Gerner and H. Rohling, "Parking lot detection with 24 ghz radar sensor," in *3rd Int. Workshop Intell. Transp.*, 2006.
- [12] H. Jung, Y. H. Cho, P. J. Yoon, and J. Kim, "Scanning laser radar-based target position designation for parking aid system," *IEEE Trans. Intell. Transp. Syst.*, vol. 9, pp. 406–424, 2008.
- [13] G. Sabiron, P. Chavent, T. Raharijaona, P. Fabiani, and F. Ruffier, "Low-speed optic-flow sensor onboard an unmanned helicopter flying outside over fields," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2013.
- [14] F. Expert and F. Ruffier, "Flying over uneven moving terrain based on optic-flow cues without any need for reference frames or accelerometers," *Bioinspiration & Biomimetics*, vol. 10, pp. , 2015.
- [15] "Peugeot - park assist." <http://www.feline208.net/dossiers/equipements/park-assist/>.
- [16] "Volkswagen - park steerig assistance." http://www.volkswagenag.com/content/vwcorp/content/en/innovation/driver_assistance/parking_steering_assistance.html.
- [17] "Toyota - intelligent park assist." <http://www.toyota.com.au/prius-v/features/audio-technology/intelligent-park-assist>.
- [18] "Nissan - around view monitor system." <http://www.nissan-global.com/EN/TECHNOLOGY/OVERVIEW/avm.html>.
- [19] "Bmw - surround view system." http://www.bmw.com/com/en/insights/technology/connecteddrive/2013/driver_assistance/intelligent_parking.html.
- [20] N. Franceschini, A. Riehle, and A. Nestour, *Facets of vision*, ch. Directionally selective motion detection by insect neurons, pp. 360–390. 1989.
- [21] S. Mafra, S. Godiot, M. Menouni, M. Boyron, F. Expert, N. Marchand, F. Ruffier, and S. Viollet, "A Bio-Inspired Analog Silicon Retina with Michaelis-Menten Auto-Adaptive Pixels Sensitive to Small and Large Changes in Light," *Optics Express*, vol. 23, pp. 5614–5635, 2015.
- [22] F. Ruffier, S. Viollet, S. Amic, and N. Franceschini, "Bio-inspired optical flow circuits for the visual guidance of micro air vehicles," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2003.
- [23] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, pp. 381–395, 1981.
- [24] R. Vidal, "Subspace Clustering," *IEEE Signal Processing Magazine*, vol. 28, pp. 52–68, 2011.
- [25] A. De Luca, G. Oriolo, and C. Samson, *Robot Motion Planning and Control*, ch. Feedback control of a nonholonomic car-like robot, vol. 229, pp. 171–253, 1998.
- [26] W. Reichardt, *Sensory Communication*, ch. Autocorrelation, a principle for the evaluation of sensory information by the central nervous system, pp. 303–317. MIT Press, 1961.